Claims:

What is claimed is:

1. A system for stopping threads in a safe state in a run-time environment comprising:

a plurality of application threads; and,

a native code interpreter which is configured stop execution of an executing thread such that the thread is stopped in a safe state.

2. The system of claim 1 wherein the system includes a virtual machine and wherein said plurality of application threads execute as part of said virtual machine.

3. The system of claim 1 wherein the system is used for the garbage collection of inactive threads in the run-time environment.

4. The system of claim 1 wherein the system is used to perform context-switching between the threads in a run-time environment.

5. The system of claim 1 wherein the native code interpreter is configured to interpret the machine code currently at the executing thread, and provide that information to the system for use in stopping the executing thread in a safe state.

6. A system for moving threads to a safe state in a run-time environment comprising:

a plurality of application threads; and, a native code interpreter which is used to allow a first or a stopping thread to roll a second or an executing thread forward such that the executing thread is stopped in a safe state.

7. The system of claim 6 wherein the system includes a virtual machine and wherein said stopping and executing threads execute as part of said virtual machine.

8. The system of claim 6 wherein the system is used for the garbage collection of inactive threads in the run-time environment.

9. The system of claim 6 wherein the system is used to perform context-switching between the threads in a run-time environment.

10. The system of claim 6 wherein the native code interpreter is configured to interpret the machine code currently at the executing thread, and provide that information to the system for use in stopping the executing thread in a safe state.

11. A system which uses native code interpretation to move threads to a safe state in a run-time environment comprising:

a first and second application threads; and,

a native code interpreter configured to allow the first thread to stop execution of the second thread and roll the second thread forward such that the thread is stopped in a safe state.

12. The system of claim 11 wherein the system includes a virtual machine and wherein said first and second application threads execute as part of said virtual machine.

13. The system of claim 11 wherein the system is used for the garbage collection of inactive threads in the run-time environment.

14. The system of claim 11 wherein the system is used to perform context-switching between the threads in a run-time environment.

15. The system of claim 11 wherein the native code interpreter is configured to interpret the machine code currently at the executing thread, and provide that information to the system for use in stopping the executing thread in a safe state.

16. A system which uses native code interpretation to stop threads in a safe state in a run-time environment, including instructions stored thereon which when executed cause the system to perform the steps of:

allowing a first thread to initially halt execution of a second thread;

using native code interpretation to determine the current state of the second thread; and,

allowing the first thread to roll forward the state of the second thread such that the second thread is stopped.

17. The system of claim 16 wherein the system includes a virtual machine and wherein said plurality of first and second threads execute as part of said virtual machine.

18. The system of claim 16 wherein the system is used for the garbage collection of inactive threads in the run-time environment.

19. The system of claim 16 wherein the system is used to perform context-switching between the threads in a run-time environment.

20. The system of claim 16 wherein the native code interpretation is performed by interpreting the machine code currently at the executing thread, and providing that information to the system for use in stopping the executing thread in a safe state.

21. A method for stopping threads in a safe state in a run-time environment, comprising the steps of:

providing a plurality of application threads; and,

providing a native code interpreter which is configured stop execution of an executing thread such that the thread is stopped in a safe state.

22. The method of claim 21 wherein the system includes a virtual machine and wherein said plurality of application threads execute as part of said virtual machine.

23. The method of claim 21 wherein the system is used for the garbage collection of inactive threads in the run-time environment.

24. The method of claim 21 wherein the system is used to perform context-switching between the threads in a run-time environment.

25. The method of claim 21 wherein the native code interpreter is configured to interpret the machine code currently at the executing thread, and provide that information to the system for use in stopping the executing thread in a safe state.

26. A method for moving threads to a safe state in a run-time environment, comprising the steps of:

providing a plurality of application threads; and,

providing a native code interpreter which is used to allow a first or a stopping thread to roll a second or an executing thread forward such that the executing thread is stopped in a safe state.

27. The method of claim 26 wherein the system includes a virtual machine and wherein said stopping and executing threads execute as part of said virtual machine.

28. The method of claim 26 wherein the system is used for the garbage collection of inactive threads in the run-time environment.

29. The method of claim 26 wherein the system is used to perform context-switching between the threads in a run-time environment.

30. The method of claim 26 wherein the native code interpreter is configured to interpret the machine code currently at the executing thread, and provide that information to the system for use in stopping the executing thread in a safe state.

31. A method which uses native code interpretation to move threads to a safe state in a run-time environment, comprising the steps of:

providing a first and second application threads; and,

providing a native code interpreter configured to allow the first thread to stop execution of the second thread and roll the second thread forward such that the thread is stopped in a safe state.

32. The method of claim 31 wherein the system includes a virtual machine and wherein said first and second application threads execute as part of said virtual machine.

33. The method of claim 31 wherein the system is used for the garbage collection of inactive threads in the run-time environment.

34. The method of claim 31 wherein the system is used to perform context-switching between the threads in a run-time environment.

35. The method of claim 31 wherein the native code interpreter is configured to interpret the machine code currently at the executing thread, and provide that information to the system for use in stopping the executing thread in a safe state.

36. A method which uses native code interpretation to stop threads in a safe state in a run-time environment, comprising the steps of:

allowing a first thread to initially halt execution of a second thread;

using native code interpretation to determine the current state of the second thread; and,

allowing the first thread to roll forward the state of the second thread such that the second thread is stopped.

37. The method of claim 36 wherein the system includes a virtual machine and wherein said plurality of first and second threads execute as part of said virtual machine.

38. The method of claim 36 wherein the system is used for the garbage collection of inactive threads in the run-time environment.

39. The method of claim 36 wherein the system is used to perform context-switching between the threads in a run-time environment.

40. The method of claim 36 wherein the native code interpretation is performed by interpreting the machine code currently at the executing thread, and providing that information to the system for use in stopping the executing thread in a safe state.

41. A computer readable medium including instructions stored thereon which when executed cause the computer to perform the steps of:

providing a plurality of application threads; and,

providing a native code interpreter which is configured stop execution of an executing thread such that the thread is stopped in a safe state.

42. The computer readable medium of claim 41 wherein the system includes a virtual machine and wherein said plurality of application threads execute as part of said virtual machine.

43. The computer readable medium of claim 41 wherein the system is used for the garbage collection of inactive threads in the run-time environment.

44. The computer readable medium of claim 41 wherein the system is used to perform context- switching between the threads in a run-time environment.

45. The computer readable medium of claim 41 wherein the native code interpreter is configured to interpret the machine code currently at the executing thread, and provide that information to the system for use in stopping the executing thread in a safe state.

46. A computer readable medium including instructions stored thereon which when executed cause the computer to perform the steps of:

providing a plurality of application threads; and,

providing a native code interpreter which is used to allow a first or a stopping thread to roll a second or an executing thread forward such that the executing thread is stopped in a safe state.

47. The computer readable medium of claim 46 wherein the system includes a virtual machine and wherein said stopping and executing threads execute as part of said virtual machine.

48. The computer readable medium of claim 46 wherein the system is used for the garbage collection of inactive threads in the run-time environment.

49. The computer readable medium of claim 46 wherein the system is used to perform context- switching between the threads in a run-time environment.

50. The computer readable medium of claim 46 wherein the native code interpreter is configured to interpret the machine code currently at the executing

thread, and provide that information to the system for use in stopping the executing thread in a safe state.

51. A computer readable medium including instructions stored thereon which when executed cause the computer to perform the steps of:

providing a first and second application threads; and,

providing a native code interpreter configured to allow the first thread to stop execution of the second thread and roll the second thread forward such that the thread is stopped in a safe state.

52. The computer readable medium of claim 51 wherein the system includes a virtual machine and wherein said first and second application threads execute as part of said virtual machine.

53. The computer readable medium of claim 51 wherein the system is used for the garbage collection of inactive threads in the run-time environment.

54. The computer readable medium of claim 51 wherein the system is used to perform context- switching between the threads in a run-time environment.

55. The computer readable medium of claim 51 wherein the native code interpreter is configured to interpret the machine code currently at the executing thread, and provide that information to the system for use in stopping the executing thread in a safe state.

56. A computer readable medium including instructions stored thereon which when executed cause the computer to perform the steps of:

allowing a first thread to initially halt execution of a second thread;

using native code interpretation to determine the current state of the second thread; and,

allowing the first thread to roll forward the state of the second thread such that the second thread is stopped.

57. The computer readable medium of claim 56 wherein the system includes a virtual machine and wherein said plurality of first and second threads execute as part of said virtual machine.

58. The computer readable medium of claim 56 wherein the system is used for the garbage collection of inactive threads in the run-time environment.

59. The computer readable medium of claim 56 wherein the system is used to perform context- switching between the threads in a run-time environment.

60. The computer readable medium of claim 56 wherein the native code interpretation is performed by interpreting the machine code currently at the executing thread, and providing that information to the system for use in stopping the executing thread in a safe state.